AMD

SAMPLE

Sparse Matrix Vector Multiplication

1 Introduction

A sparse matrix is composed mostly of zeroes. Sparse matrices are manifested in physical phenomena described by partial differential equations (PDEs). Figure 1 shows a visualization of a sparse-matrix obtained in a structural problem from the University of Florida sparse-matrix collection [1]. A fast solution for sparse matrix-vector multiplication on the GPU is critical for writing high performance PDE solvers. We provide a Brook+ sample for fast sparse matrix-vector multiplication.

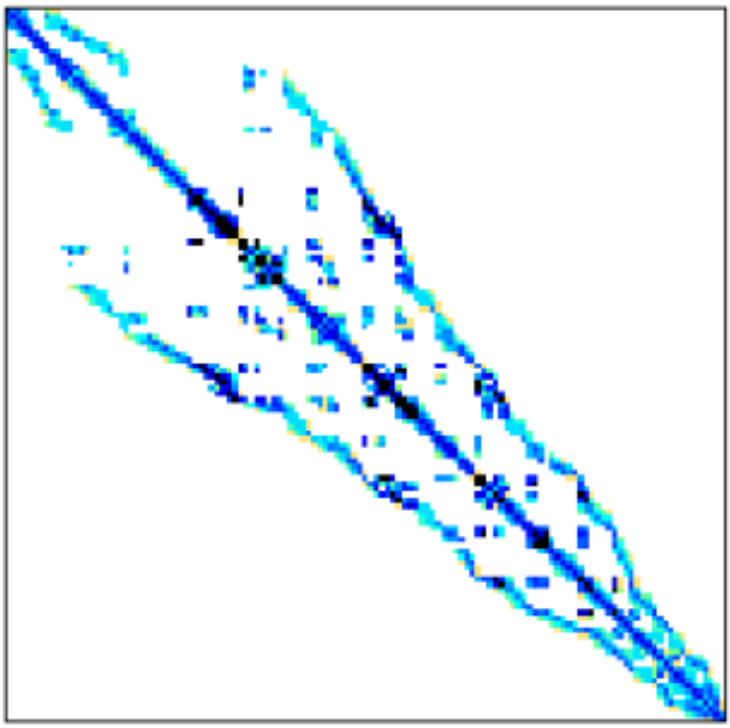

Figure 1 Sparse-Matrix Obtained in a Structural Problem; Non-zero Elements are Colored (Source: University of Florida Sparse-Matrix Collection)

2 Compressed Sparse Row (CSR) Format

Storing a sparse-matrix in regular row-major or column-major format is wasteful because most of the memory is occupied by zero entries. The Brook+ sample supports the Compressed Sparse Row (CSR) format for sparse-matrix storage. This stores only the non-zero elements of the matrix and is specified by three arrays:

values: contains the non-zero elements of the matrix. Elements are read off each row of the matrix and stored in the array in order.

column number: the i^{th} entry in this array specifies the column number of the i^{th} entry in the values array.

row index: the i^{th} entry in this array specifies the index in the values array of the first non-zero element of the i^{th} row in the matrix.

See Figure 2 for an illustration of the compressed row format.

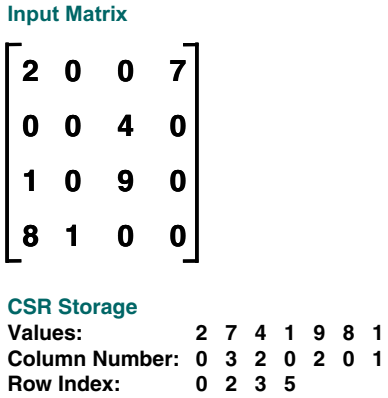

Figure 2 CSR Storage for a 4 x 4 Matrix (Assuming Indices Start at 0)

3 ITPACK Padded Compressed Row Format

For the Brook+ SIMD implementation of sparse-matrix multiplication we convert from the CSR format to the ITPACK Padded Compressed Row format because it is more suitable for streaming applications.

We enforce the constraint that each row of the matrix can have at most **w** nonzero elements. Now, we store all non-zero elements in a vector of length m x w, where m is the number of rows in the original matrix. If a row has less than w non-zero elements, it is padded with zeros. As before, we also store column numbers for each non-zero element. We store a zero for the column number of any zero entry that occurs as a result of padding. This does not affect the result, but it can have an insignificant affect on performance. Figure 3 shows ITPACK storage for our example 4x4 matrix.

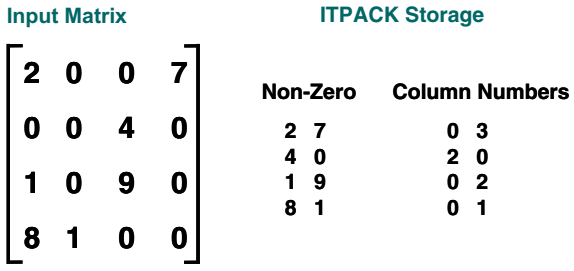

Figure 3 ITPACK Padded Compressed Row Format storage for a 4x4 matrix (assuming indices start at 0); w = 2

4 BROOK+ SIMD Sparse-Matrix Vector Multiplication

We want to multiply a sparse-matrix **A** of dimensions m x n represented in the ITPACK padded compressed row format with a vector **x** of length n. This is achieved in three simple steps.

1. **Gather**: The elements of vector **x** are gathered into a stream **t**, the access pattern being specified by the column numbers of the non-zero elements of A. This arranges them in the order in which multiplication is to be done.
2. **Multiply**: The gathered x vector are component-wise multiplied with the nonzeros of A.
3. **Sum**: Reduction via the sum operator is performed on the stream obtained as a result of two. This accumulates the sums across rows and gives us the answer.

Figure 4 shows this process for an example input vector, x.

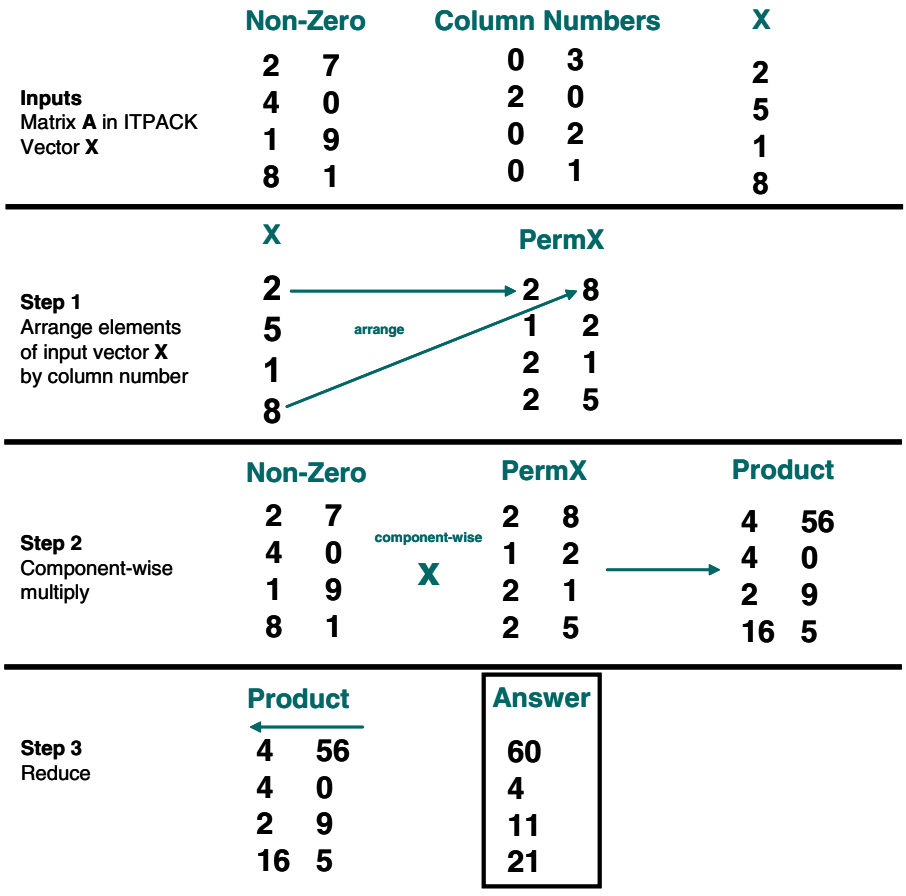

Figure 4 Brook+ Sparse Matrix-Vector Multiplication

5 Performance

The performance of Brook+ Sparse Matrix-Vector Multiplication was tested on a system with the AMD Athlon™ 64 X2 Dual Core Processor CPU and an ATI Radeon™ HD 4800 Series GPU. The Figure 5 shows a comparison of the time taken to compute Sparse Matrix-Vector products for matrices of varying sizes on the stream processor versus the CPU.

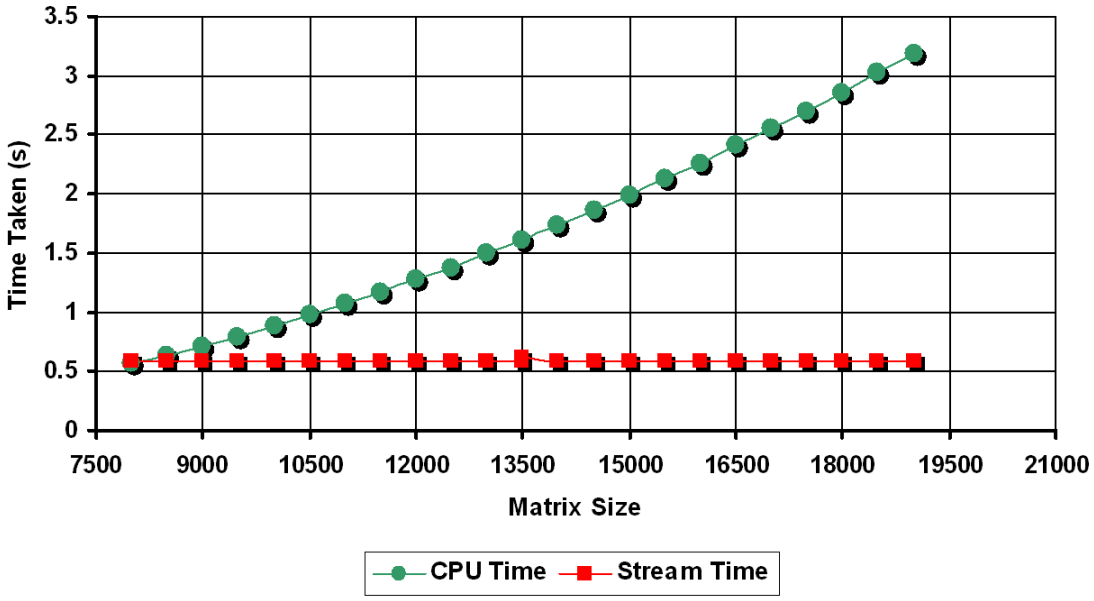

Figure 5 Performance of Sparse Matrix-Vector Multiplication: Stream Processor vs CPU

It also shows speedup achieved over the reference CPU implementation. The AMD Radeon™ 4800 Series can achieve speedups of over 25 for this sample implementation.

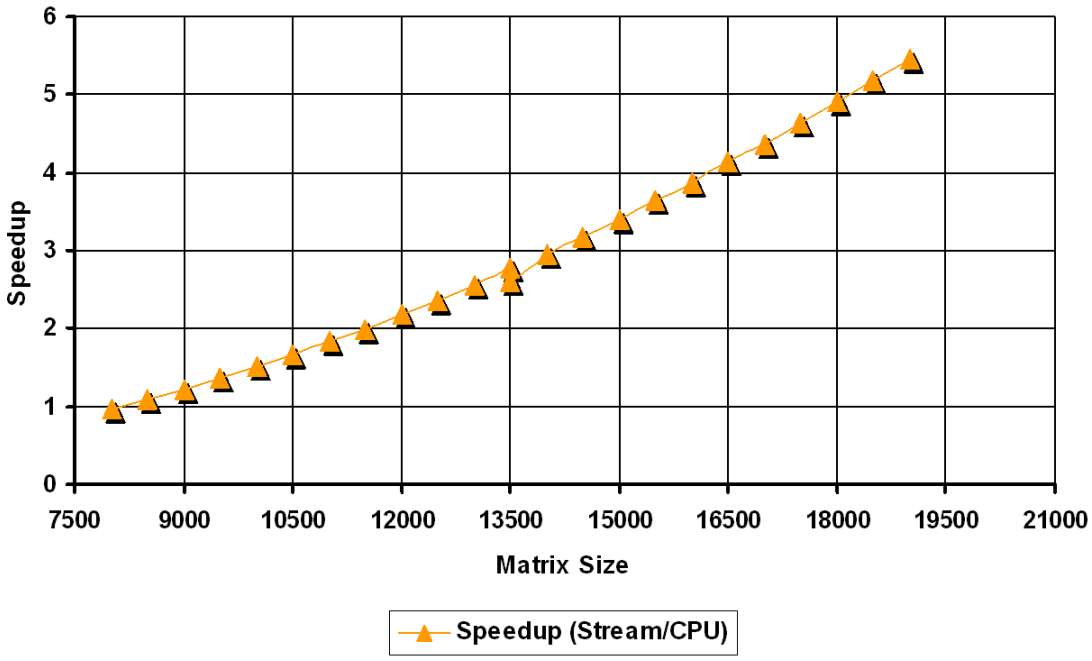

Figure 6 Speedup of Sparse Matrix-Vector Multiplication: Stream vs CPU

6 References

T. Davis, The University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices, submitted to *ACM Trans. on Mathematical Software*. See also: *NA Digest*, vol. 92, no. 42, October 16, 1994; *NA Digest*, vol. 96, no. 28, July 23, 1996; and *NA Digest*, vol. 97, no. 23, June 7, 1997.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For Stream Computing:
URL: http://ati.amd.com/technology/streamcomputing
Questions: streamcomputing@amd.com
Developing: streamdeveloper@amd.com
Forum: http://forums.amd.com/devforum/categories.cfm?catid=38